



AWS 시작 복원력 기준

AWS 권장 가이드



AWS 권장 가이드: AWS 시작 복원력 기준

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

소개	1
지침 범위	1
공동 책임 모델	2
1단계: 목표 설정	3
2단계: 설계 및 구현	5
3단계: 평가 및 테스트	7
4단계: 운영	8
5단계: 대응 및 학습	10
다음 단계	11
리소스	12
AWS Well-Architected 프레임워크	12
AWS 권장 가이드	12
기타 AWS 리소스	12
기여자	13
작성	13
검토	13
기술 작성	13
문서 이력	14
.....	xv

AWS 시작 복원력 기준

Amazon Web Services([기여자](#))

2026년 3월([문서 기록](#))

모든 스타트업 설립자는 긴장을 알고 있습니다. 팀이 새로운 기능을 제공하기 위해 경쟁하는 반면 고객은 모든 릴리스가 제대로 작동하기를 기대합니다. 어젯밤의 운영 중단에 대한 알림을 받고, 고객 불만 제기가 진행 중이며, 로드맵에 이미 약속된 기능이 가득 차 있습니다. 이는 안정적인 서비스를 유지해야 하는 필요성과 혁신에 대한 압력의 균형을 맞추는 스타트업의 일상적인 현실입니다.

새로운 기능의 우선 순위를 지정하는 본능은 자연스럽습니다. 투자자는 성장을 원하고, 고객은 더 많은 기능을 요구하며, 경쟁자는 아직 멈춰 있지 않습니다. 그러나 경험에 따르면 가장 혁신적인 스타트업도 복원력 문제를 겪을 수 있습니다. 서비스 중단은 수익 손실을 의미할 뿐만 아니라 고객과 구축하기 위해 노력한 신뢰를 약화시킵니다. 실제로 바위처럼 안정적인 신뢰성은 혼잡한 시장에서 무음 차별화 요소가 될 수 있습니다.

시작 시 복원력을 구축한다고 해서 속도가 느려지거나 대규모 인프라 투자가 필요한 것은 아닙니다. 나중에 문제를 방지하기 위해 조기에 스마트한 선택을 하는 것입니다. 집을 짓는 것처럼 생각하세요. 처음부터 견고한 기반을 마련하는 것이 모든 것이 구축되면 구조적 문제를 해결하는 것보다 훨씬 쉽습니다. 초기 아키텍처 및 운영에 복원력 있는 관행을 적용하면 경쟁 우위를 확보할 수 있습니다. 단순히 제품을 제공하는 것이 아니라 고객 충성도를 높이는 신뢰할 수 있는 경험을 제공하는 것입니다.

AWS 시작 복원력 기준(AWS SRB)은 이 문제에 직면한 팀을 위해 특별히 생성되었습니다. 시작을 특별하게 만드는 속도를 유지하면서 시스템에 신뢰성을 구축할 수 있는 실용적인 경로를 제공합니다. 프로세스가 많거나 엔터프라이즈 규모의 복잡성이 없으며 비즈니스에 따라 증가하는 실용적인 패턴만 있습니다.

이 가이드에서는 시작 환경에서 AWS SRB를 구현하는 방법을 설명합니다. 이는 복원력에 실제로 중요한 요소, 제한된 리소스에 집중해야 하는 위치, 회사가 성장함에 따라 확장되는 사례를 구축하는 방법을 식별하는 데 도움이 됩니다.

지침 범위

스타트업 복원력을 구축하는 실제 단계를 탐색할 때는 이 지침이 조직의 광범위한 성장 궤도에 맞는 위치를 이해하는 것이 중요합니다. AWS Startup Resiliency Baseline은 [복원력 수명 주기 프레임워크](#)와 일치합니다. 첫 번째 애플리케이션을 배포할 때 개발 오버헤드를 최소화하면서 기본 복원력 조치를 구

현하는 데 도움이 되도록 설계된 초기 로드맵 역할을 합니다 AWS. 이를 중단으로부터 효과적으로 복구할 수 있는 신뢰할 수 있는 시스템을 구축하기 위한 스타터 키트로 생각하세요.

[AWS Well-Architected Framework](#)와 이 복원력 지침은 안전하고 신뢰할 수 있는 인프라를 구축하는 데 도움이 되도록 서로 보완합니다. 이 가이드에서는 Well AWS -Architected Framework가 더 광범위한 아키텍처 모범 사례를 제공하는 동안 복원력 사례를 자세히 살펴봅니다. [AWS Well-Architected Tool](#)의 를 사용하여 이러한 사례를 기준으로 아키텍처를 평가할 수 있습니다 AWS 계정.

공동 책임 모델

클라우드에서 복원력 있는 시스템을 구축하는 중요한 측면, 즉 AWS 와 스타트업 간의 파트너십을 명확히 하는 것이 중요합니다. 클라우드의 복원력은 [공동 책임 모델에서](#) 작동 AWS 하며, 팀은 각각 시스템 복원력을 보장하는 데 있어 고유한 역할을 합니다.

AWS 는 서비스를 지원하는 기본 클라우드 인프라의 복원력에 대한 책임을 집니다. 이를 애플리케이션을 빌드하는 기반으로 생각하십시오. 여기에는 데이터 센터, 네트워킹 구성 요소 및 아키텍처에서 AWS 서비스 사용하는 코어의 복원력 유지가 포함됩니다.

스타트업의 책임은 이 기반에서 구축하고 배포하는 시스템의 복원력에 중점을 둡니다. 이 가이드의 권장 사항은 책임 영역에 속합니다. 이러한 관행을 신중하게 구현하면 안정적인 AWS 인프라를 사용하고 강력한 복구 기능을 통합하는 애플리케이션을 만들 수 있습니다.

이 모델은 복원력을 독립적으로 구축하지 않음을 의미합니다. 고객과 비즈니스 운영에 직접적인 영향을 미치는 측면에 노력을 집중하면서 입증되고 신뢰할 수 있는 기반을 구축하고 있습니다. 이 가이드에서는 신뢰할 수 있는 AWS 인프라를 사용하는 동시에 실제 조치를 구현하여 애플리케이션이 중단으로부터 정상적으로 복구되도록 하여 이러한 공동 책임 모델을 최대한 활용하는 방법을 살펴봅니다.

1단계: 목표 설정

주요 제품 출시 전에 팀이 마지막 스프린트에 있다고 가정해 보겠습니다. 새로운 기능은 획기적인 기능이며 투자자의 기대가 고조되고 있습니다. 그런 다음 일상적인 배포 중에 코어 서비스가 중단됩니다. 고객 불만 제기로 인해 이메일이 넘치면 두 가지 질문, 즉 얼마나 오래 오프라인 상태가 될 수 있는지가 명확해집니다. 손실할 수 있는 데이터는 무엇입니까?

모든 것이 잘 작동하기를 바라는 것은 좋은 전략이 아닙니다. 복원력이 가장 중요한 부분과 중요하지 않은 부분을 결정하는 체계적인 방법이 필요합니다. 여기서 비즈니스 영향 분석(BIA)이 중요해집니다. 복원력에 투자할 위치에 대해 정보에 입각한 결정을 내리는 데 도움이 됩니다. BIA는 시스템의 어떤 부분이 실제로 강력한 신뢰성을 필요로 하고 어떤 부분이 어느 정도 유연성을 허용할 수 있는지 이해하는 데 도움이 됩니다.

먼저 핵심 사용자 여정을 매핑합니다. 각 항목에 대해 다음과 같이 자문해 보세요.

- 이것이 중단되면 어떤 영향이 있습니까?
- 서비스를 얼마나 빨리 복원해야 합니까?
- 보호에 중요한 데이터는 무엇입니까?

이는 단순한 기술적 연습이 아니라 신뢰성 문제의 비즈니스 영향을 이해하는 데 도움이 됩니다. 수익 손실은 시작에 불과합니다. 운영 중단으로 인해 고객 신뢰가 약화되거나, 규제 요구 사항을 위반하거나, 경쟁자에게 경쟁 우위를 제공하는 방법을 고려합니다.

이 분석에서는 각 사용자 여정에 대해 복구 시간 목표(RTO)와 복구 시점 목표(RPO)라는 두 가지 중요한 숫자를 도출합니다. RTO는 해당 여정을 얼마나 빨리 복원해야 하는지 정의합니다. RPO는 고객이 허용할 수 있는 데이터 손실의 양을 정의합니다. 그런 다음 이러한 비즈니스 기반 대상은 시스템의 모든 부분을 과도하게 엔지니어링하지 않고도 선택하는 구성 요소와 이를 설계하는 방법을 안내합니다.

이 접근 방식의 장점은 제한된 리소스를 가장 중요한 위치에 집중하는 데 도움이 된다는 것입니다. 코어 트랜잭션 처리에는 거의 즉각적인 복구와 제로 데이터 손실이 필요하지만 권장 엔진은 더 긴 가동 중지 시간을 허용할 수 있습니다. 명확한 목표를 설정하면 복원력을 전략적으로 구축하면서 빠른 기능 개발을 계속할 수 있는 프레임워크를 생성할 수 있습니다.

이러한 목표를 명확하게 문서화합니다. 엔지니어링 팀만을 위한 것이 아닙니다. 기업 고객에게 홍보하거나 투자자와 기술 실사를 진행할 때 설명서는 비즈니스 연속성에 대해 비판적으로 생각했음을 보여줍니다.

이러한 대상은 스타트업이 성장함에 따라 진화합니다. 처음 수천 명의 사용자의 복원력 요구 사항은 첫 번째 엔터프라이즈 클라이언트의 복원력 요구 사항과 다릅니다. 오늘 현실적으로 달성할 수 있는 목표부터 시작하되, 규모 조정에 따라 어떻게 축소할지 계획합니다.

이 가이드에서는 이러한 목표를 충족하는 복원력 조치를 구현하는 방법을 살펴봅니다. 이러한 대상을 설정하는 것이 중요한 첫 번째 단계입니다. 혁신과 안정성 간의 지속적인 긴장을 극복하기 위한 나침반입니다. 이를 통해 고객에게 신뢰할 수 있는 가치를 제공하는 시스템을 구축할 수 있습니다.

2단계: 설계 및 구현

이 섹션에서는 복원력 목표를 실현하는 방법에 대해 설명합니다. 비즈니스에 가장 중요한 것을 매핑했으므로 이제 구축할 차례입니다. 혁신을 늦추지 않고 복원력을 구축하려면 어떻게 해야 하나요?

AWS 관리형 서비스를 복원력 바로 가기로 생각하세요. 인프라를 유지 관리하는 귀중한 엔지니어링 시간을 소모하는 대신 중복성을 처리하는 서비스를 사용하세요. 예를 들어 [Amazon Simple Storage Service\(Amazon S3\)](#). 내구성을 AWS 리전 위해에 데이터의 여러 복사본을 자동으로 저장합니다. 추가 코드나 야간 페이지 업무는 필요하지 않습니다.

핵심 애플리케이션 구성 요소는 어떤가요? 스마트한 선택은 팀의 영향을 배가할 수 있습니다. 서비스의 백본인 데이터베이스를 생각해 보세요. 자체 복제 시스템을 구축하는 대신 장애 조치를 자동으로 처리하는 [Amazon Aurora](#)를 사용하는 것이 좋습니다. 이러한 기능은 비용이 더 많이 들 수 있지만 팀의 집중력을 인프라 유지 관리에서 비즈니스 문제 해결로 전환합니다. 이 비용은 더 빠른 기능 제공과 중단 시 수익 손실을 방지하여 상쇄할 수 있습니다.

스타트업이 사용자 지정 솔루션을 구축해야 하는 경우가 있습니다. 이것이 혁신적인 스타트업의 특성입니다. 이렇게 할 때는 단순하지만 스마트하게 유지하세요. [Elastic Load Balancing](#) 및 [Amazon EC2 Auto Scaling 그룹](#)을 사용하여 여러 가용 영역에 애플리케이션을 분산합니다. 가용 영역 하나가 실패하더라도 기존 트래픽을 처리하도록 Auto Scaling 그룹 최소 용량을 설정합니다. 이를 통해 복잡한 아키텍처 패턴 없이 현지화된 장애에 대한 복원력을 제공합니다. 스타트업이 성장하고 고객이 더 높은 복원력을 요구함에 따라 더 정교한 접근 방식으로 발전할 수 있습니다.

프로덕션 및 개발 환경을 별도의 계정에 보관하는 것이 좋습니다 AWS 계정. 빠르게 이동할 때 믹싱하고 싶지만이 경계는 안전망입니다. 이를 통해 의미 있는 실험이 프로덕션 서비스를 중단하는 것을 방지할 수 있습니다. 이를 "빠르게 움직이고 사물을 깨는" 개발 문화에 대한 보험이라고 생각하십시오. 개발 과정에서 사물을 깨고 프로덕션을 안정적으로 유지합니다.

애플리케이션이 타사 서비스에 의존하는 경우 장애를 계획합니다. 결제 프로세서에 문제가 있는 경우 시스템에서 정상적으로 처리할 수 있습니까? 간단한 회로 차단기와 대체 옵션을 구축합니다. 오류 메시지를 표시하는 대신 이러한 트랜잭션을 대기열에 넣을 수 있습니다. 고객은 완벽하게 작동하지 않더라도 계속 작동했다는 점을 감사할 것입니다.

빌드할 때 문서화하되 실용적으로 유지합니다. 주요 결정의 배경이 되는 이유를 기록하는 데 집중하고 간단한 복구 플레이북을 생성합니다. 인시던트가 발생할 때 이를 준비하는 것이 중요합니다.

완벽한 복원력을 위해 구축하는 것이 아니라 적절한 복원력을 위해 구축하는 것입니다. 과다 엔지니어링 복원력에 소비되는 매시간은 고객이 요청하는 기능에 소비되지 않는 한 시간입니다. AWS 관리형

서비스를 기반으로 사용하고, 가장 중요한 곳에 대상 복원력을 추가하고, 비즈니스가 성장함에 따라 복원력을 확장할 수 있는 명확한 경로를 만듭니다.

다음 장에서는 엔지니어링 리소스를 소모하지 않고 이러한 설계 선택 사항을 검증하는 방법을 설명합니다. 스타트업의 경우 테스트는 적절한 리프트와 애플리케이션 복원력에 대한 스마트 투자여야 합니다.

3단계: 평가 및 테스트

복원력이 뛰어난 기반을 구축했지만 실제로 작동하는지 어떻게 알 수 있나요? 제품 시장 적합성을 입증하기 위해 레이싱할 때 복원력을 테스트하는 것은 고급스러울 수 있습니다. 그러나 기능 개발을 방해하지 않고 이를 수행할 수 있는 스마트한 방법이 있습니다. 이 장에서는 스타트업의 속도에 맞는 린하고 실용적인 테스트를 설명합니다.

로 시작하여 초기 아키텍처 평가 도구로 [AWS Resilience Hub](#) 생각하십시오. 아키텍처의 복원력 기반에 대한 유용한 기준 검토를 제공합니다. 이를 통해 일반적인 구성 패턴과 잠재적 단일 장애 지점을 확인하여 기본 인프라 설정이 복구 목표에 부합하는지 평가할 수 있습니다. 여러 가용 영역 구성 누락 또는 불완전한 백업 정책과 같은 명백한 격차에 플래그를 지정할 수 있습니다. Resilience Hub는 사려 깊은 아키텍처 검토와 중요한 경로의 대상 테스트를 보완하지만 대체하지는 않습니다.

문서화된 복구 목표를 검증하려면 개발 환경에서 [AWS Backup](#)에서 월별 복원 테스트를 예약합니다. 엔지니어링 시간이 필요하지만 실제 인시던트 발생 시 백업 검색이 작동하지 않는 것보다 저렴할 수 있습니다. 단위 테스트 또는 코드 검토 실행과 같은 정기적인 개발 주기의 일부로 만듭니다. 목표는 완벽하지 않고 필요할 때 복구할 수 있다는 확신입니다.

스타트업이 성장하고 고객의 의존도가 높아지면 테스트 게임의 수준을 점진적으로 높일 수 있습니다. 새 기능을 배포할 때 파이프라인에 기본 복원력 검사를 포함합니다. 를 사용하여 간단한 카오스 실험을 시도합니다 [AWS Fault Injection Service](#). 사전 프로덕션 환경에서 시작하고 작게 시작합니다. 프로덕션 실험을 고려하기 전에 애플리케이션이 개발 중에 지연된 API 응답을 처리하는 방법을 테스트합니다. 신뢰도가 높아지면 이러한 테스트를 점진적으로 확장하되 항상 사전 프로덕션에서 먼저 검증합니다. 스타트업의 경우 프로덕션 환경에서 사물을 깨는 것은 의도적으로 수행하지 않으면 충분히 위험합니다.

키는 밸런스입니다. 테스트에 소비되는 매시간은 새로운 기능을 구축하는 데 소비되지 않는 한 시간입니다. 그러나 몇 가지 전략적 테스트는 고객의 신뢰를 잃는 종류의 종단을 방지할 수 있습니다. 에서 제공하는 자동화된 도구를 사용하여 과중한 작업을 AWS 수행하고 고객에게 가장 중요한 테스트에 집중합니다. 이를 통해 혁신을 늦추지 않고 애플리케이션의 복원력에 대한 신뢰를 구축할 수 있습니다.

다음 장에서는 시작 규모에 따라이 기반을 발전시키는 방법을 살펴봅니다.

4단계: 운영

복원력이 뛰어난 애플리케이션을 빌드하고 테스트했습니다. 이제 일상적인 현실이 계속 실행되고 있습니다. 그러나 시작 시 모든 작업을 볼 수 없으며 시도해서는 안 됩니다. 핵심은 지표를 너무 많이 제공하거나 팀을 과도하게 부담하지 않고 중요한 사항에 주의를 기울이는 것입니다.

고객 관점에서 시작합니다. [Amazon CloudWatch Synthetics](#) canary는 자동화된 고객 역할을 합니다. 중요한 사용자 여정을 지속적으로 테스트합니다. 특히 바쁜 시간에 로그인하거나, 테스트 계정을 사용하여 구매를 시뮬레이션하거나, 주요 기능에 액세스하도록 합니다. 이를 통해 고객 경험을 이해하고 실제 사용자가 문제를 해결하기 전에 문제를 포착할 수 있습니다. canary가 실패하면 고객의 관점에서 문제가 있음을 즉시 알 수 있습니다.

지원 인프라를 집중적으로 모니터링하여이 기반을 구축합니다. 문제가 있음을 알려주는 신호는 무엇입니까? [Amazon CloudWatch](#)를 사용하면 이러한 징후를 추적하는 대시보드를 구축할 수 있습니다. 기술 지표만 모니터링하지 말고 비즈니스에 미치는 영향과 연결하세요. 예를 들어 CPU 사용량이 높을수록 카나리아로 추적하는 고객 경험이 저하될 수 있기 때문입니다.

실용적인 접근 방식으로 모니터링을 고객 여정에 매핑합니다. 서비스형 소프트웨어(SaaS) 플랫폼을 실행하는 경우 API 응답 시간, 인증 성공률 및 핵심 기능 가용성에 신경을 쓸 것입니다. 이러한 지표가 드리프트되는 시기를 알려주는 알림을 설정합니다. 그러나 선택 사항이어야 합니다. 모든 알림에는 조치가 필요합니다. “아마도 아무것도 아니기 때문에” 팀이 알림을 무시하기 시작하는 경우 너무 많이 설정했거나 잘못된 지표를 추적하고 있는 것입니다.

팀이 이미 사용하는 도구를 통해 이러한 알림을 라우팅합니다. 엔지니어가 특정 메시징 애플리케이션에 거주하는 경우 여기에 알림을 보냅니다. 목표는 새 프로세스를 생성하지 않고 빠르게 인식하는 것입니다. 알림이 실행되면 팀은 알림의 의미와 이에 대해 수행할 작업을 정확히 알아야 합니다.

운영 설명서를 간결하고 실용적으로 유지하세요. 런북을 코드와 함께 버전 관리에 저장하되, 소설이 아님을 기억하세요. 문제가 발생하면 팀에 명확하고 실행 가능한 단계가 필요합니다. 각 알림은 해당 실행서에 연결되어야 하며, 각 실행서는 다음 세 가지 질문에 답해야 합니다.

- 무엇이 깨졌나요?
- 이 선택이 중요한 이유
- 어떻게 해결해야 할까요?

간단한 인시던트 관리 프로세스를 구현합니다. 복잡한 프레임워크가 필요하지 않습니다. 인시던트를 구성하는 항목과 상황이 에스컬레이션될 때 호출할 대상에 대한 명확한 정의만 있으면 됩니다. 인시던트 로그는 애플리케이션의 복원력을 개선하는 데 도움이 되므로 보관합니다.

핵심은 경계와 오버헤드 사이의 스위트 스팟을 찾는 것입니다. AWS 도구를 사용하여 할 수 있는 일을 자동화하고, 고객에게 영향을 미치는 지표를 모니터링하는 데 집중하고, 성장에 따라 프로세스를 충분히 가볍게 발전시킬 수 있습니다.

다음 장에서는 스타트업을 특별하게 만드는 속도와 혁신을 희생하지 않고 복원력 사고방식을 조성하는 방법을 살펴봅니다. 결국 복원력은 기술에 관한 것만큼이나 사람에 대한 것입니다.

5단계: 대응 및 학습

시작을 실행할 때 복잡한 사후 프로세스가 팀의 속도를 늦출 수 있습니다. 이 장에서는 인시던트를 관료적인 연습으로 전환하지 않고 인시던트에서 학습하는 방법을 살펴봅니다.

인시던트 학습을 기존 리듬에 통합합니다. 팀에 이미 정기 회의가 있는 경우 10분을 사용하여 최근 인시던트를 논의합니다. 다음과 같은 실용적인 질문에 집중합니다.

- 실행서가 도움이 되었나요?
- 알림이 적시에 발생했나요?
- AWS 관리형 서비스가 이를 방지할 수 있었을까요?

비난이 아닌 행동에 집중하세요. 스타트업에서는 완벽한 시스템을 구축하지 않고 문제가 발생할 때마다 개선되는 시스템을 구축합니다.

티케팅 시스템을 사용하여 인시던트를 추적할 수 있습니다. 특수 도구가 필요하지 않습니다. 인시던트 타임라인, 고객 영향, 수행한 복구 단계 및 학습한 내용을 포함하는 간단한 템플릿을 생성합니다. 이 캠프를 적극적으로 사용하면 기관 메모리가 됩니다. 온보딩 중에 발생한 과거 인시던트를 검토하여 신규 엔지니어의 속도를 높이세요. 유사한 시스템을 설계할 때 아키텍처 검토에서 참조하세요. 게임 데이터 가져와 실제 이벤트를 기반으로 현실적인 실패 시나리오를 생성합니다. 템플릿은 발생한 일을 캡처하고 정기적으로 사용하면 이를 조직 학습으로 변환합니다.

스타트업이 성장함에 따라 패턴이 나타납니다. 특정 구성 요소가 더 자주 실패하거나 특정 유형의 변경으로 인해 문제가 발생할 수 있습니다. 이러한 패턴을 사용하여 복원력 투자를 안내합니다. 데이터베이스 장애 조치로 인해 문제가 발생하는 경우 여러 가용 영역 설정을 개선하는 것이 좋습니다. 타사 서비스 중단이 일반적인 테마인 경우 회로 차단기를 개선하는 것이 좋습니다.

가능한 모든 장애를 방지하는 것이 목표는 아닙니다. 이는 불가능하며 속도가 너무 느려집니다. 목표는 빠르게 학습하고, 빠르게 적응하고, 빠르게 성장하는 동안 애플리케이션을 충분히 안정적으로 유지하는 것입니다. 각 인시던트를 사용하여 시스템을 약간 더 복원력이 뛰어나고, 팀이 조금 더 지식이 풍부하며, 고객이 서비스에 조금 더 확신할 수 있는 기회로 삼습니다. 스타트업의 경우 속도 및 학습 비트 완벽성. 혁신을 늦추지 않고 인시던트에서 학습하는 데 도움이 되는 경량 프로세스를 생성합니다. 가장 좋은 복원력 방법은 팀이 실제로 사용하는 것입니다.

스타트업을 위한 다음 단계

밤늦은 배포로 인해 하트 레이스가 이루어졌다는 사실을 기억하시나요? 아니면 주요 고객이 복원력 측정에 대해 질문한 순간인가요? 스타트업의 복원력 여정은 이러한 순간을 없애는 것이 아니라 자신 있게 직면하는 것입니다.

이 가이드에서는 현실적인 복구 목표 설정, AWS 관리형 서비스 사용, 린 테스트 구현, 중요한 사항 모니터링, 인시던트에서 학습 등 복원력에 대한 실용적인 경로를 설명합니다. 이 접근 방식은 함께 성장하기 때문에 스타트업에게 유용합니다. 현재 스타트업이 문제에서 복구하는 데 도움이 되는 동일한 프레임워크는 미래의 엔터프라이즈 고객에게 서비스를 제공하기 위한 기반을 구축합니다.

복원력을 제품 로드맵처럼 생각하십시오. 한 번에 모든 것을 구축하지는 않습니다. 핵심 서비스 보호부터 시작합니다. 중요한 고객 여정에 대한 모니터링을 추가합니다. 명백한 문제를 포착하는 기본 테스트를 구현합니다. 학습한 내용을 문서화합니다. 각 단계는 관리 가능하고 실용적이며, 가장 중요한 것은 특성 배송을 중단하지 않는다는 것입니다.

스타트업이 성장함에 따라 복원력이 진화해야 합니다. 엔터프라이즈 고객은 더 어려운 질문을 할 수 있습니다. 피크 트래픽은 새로운 최고치에 도달할 수 있습니다. 국제적 확장으로 인해 새로운 과제가 발생합니다. 파운데이션에 복원력을 구축했으므로 적응할 준비가 되었습니다.

스타트업의 목표는 완벽한 가동 시간이 아니거나 인시던트가 없는 것입니다. 목표는 스타트업을 특별하게 만드는 속도를 유지하면서 고객의 신뢰를 얻을 수 있을 만큼 신뢰할 수 있는 시스템을 구축하는 것입니다. 이 접근 방식을 따르면 완벽한 시스템을 만드는 것보다 더 가치 있는 것을 만들 수 있습니다. 스타트업의 야망에 발맞춰 각 챌린지에서 학습하여 매일 더 나은 시스템을 만들 수 있습니다.

복원력 여정은 끝나지 않습니다. 배송하는 모든 기능, 수주하는 모든 고객, 처리하는 모든 인시던트에 따라 진화합니다. 이 가이드는 자신감 있고 실용적인 단계를 진행하는 데 도움이 되는 프레임워크를 구축하는 데 도움이 됩니다. 결국 시작 성공은 복원력과 속도 중에서 선택하는 것이 아닙니다. 둘 다 제공할 수 있는 스마트한 방법을 찾는 것입니다.

리소스

AWS Well-Architected 프레임워크

- [안정성 원칙](#)
 - [REL05-BP01 관련 하드 종속성을 소프트웨어 종속성으로 변환하는 단계적 성능 저하 구현](#)
 - [REL06-BP01 워크로드의 모든 구성 요소 모니터링](#)
 - [REL06-BP03 알림 전송\(실시간 처리 및 경보\)](#)
 - [REL08-BP03 배포의 일부로 복원력 테스트 통합](#)
- [복원력에 대한 공동 책임 모델](#)

AWS 권장 가이드

- [의 백업 및 복구 접근 방식 AWS](#)
- [복원력 수명 주기 프레임워크: 복원력 개선에 대한 지속적인 접근 방식](#)

기타 AWS 리소스

- [AWS 장애 격리 경계](#)(AWS 백서)
- [클라우드 애플리케이션을 위한 RPO 및 RTO 대상 설정](#)(AWS 블로그 게시물)
- [의 배포 옵션 개요 AWS](#)(AWS 백서)

기여자

다음은 이 가이드에 기여한 사람입니다.

작성

- Dylan McCarroll, Associate Solutions Architect, AWS
- Igor Fil, Startup Solutions Architect, AWS
- Parth Shah, Senior Solutions Architect, AWS
- Vrajesh Prajapati, 솔루션 아키텍트, AWS

검토

- " Richey, 수석 기술자, AWS
- Bruno Emer, 수석 기술자, AWS

기술 작성

- GxP AbouHarb, Senior Technical Writer, AWS

문서 이력

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
중요 업데이트	AWS 서비스 전체의 권장 사항 및 사용을 업데이트했습니다.	2026년 3월 6일
최초 게시	—	2024년 1월 24일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.